AFCRL-63-137

63 4 2

Scientific Report No. 6
Contract AF19(604)-7493
March 1963

• pml • applied • mathematics •

Decoding Rules for Certain Product Codes

by

L. Calabi
H. G. Haefeli (Subcontractor)

Air Force Cambridge Research Laboratories
Office of Aerospace Research
United States Air Force
Bedford, Massachusetts

PARKE MATHEMATICAL LABORATORIES, Inc.
One River Road • Carlisle, Massachusetts

• pml • applied • mathematics •

AFCRL -63-137

Scientific Report No. 6
Contract AF19(604)-7493
March 1963

• applied • mathematics •

Decoding Rules for Certain Product Codes

by

L. Calabi
H.G. Haefeli (Subcontractor)

Air Force Cambridge Research Laboratories
Office of Aerospace Research
United States Air Force
Bedford Massachusetts

PARKE MATHEMATICAL LABORATORIES, Inc.
One River Road • Carlisle, Massachusetts

• applied • mathematics •

PARKE MATHEMATICAL LABORATORIES, INCORPORATED
ONE RIVER ROAD • CARLISLE, MASSACHUSETTS

Requests for additional copies by Agencies of the Department of Defense, their contractors, and other Government agencies should be directed to the:

DEFENSE DOCUMENTATION CENTER (D.D.C.)
ARLINGTON HALL STATION
ARLINGTON 12, VIRGINIA

Department of Defense contractors must be established for ASTIA services or have their 'need-to-know' certified by the cognizant military agency of their project or contract.

All other persons and organizations should apply to the:

U.S. DEPARTMENT OF COMMERCE
OFFICE OF TECHNICAL SERVICES
WASHINGTON 25, D.C.

A limited number of copies are also available by writing to:

PARKE MATHEMATICAL LABORATORIES, INC.
ONE RIVER ROAD
CARLISLE, MASSACHUSETTS

PARKE MATHEMATICAL LABORATORIES, INCORPORATED
ONE RIVER ROAD • CARLISLE, MASSACHUSETTS

Abstract

Decoding procedures are formulated, and their performance established, for products of binary group codes of even minimal weight and for Hobbs' codes of any dimension.

Table of Contents

Introduction

In our effort to discover decoding procedures for Kautz codes K_γ^δ, we have found one for the particular class of Hobbs' codes K_1^δ, based on the product structure of these codes. Our procedure cannot, then, be immediately extended to codes K_γ^δ, since these are not necessarily products, if $\gamma > 1$; but it was easy to deduce decoding rules for arbitrary products AB (somehow related to considerations of [1]). These are set forth and studied in Chapter I below, where we have restricted ourselves to consider factor codes A, B for which our decoding procedure has substantial performance. More specifically we consider only binary group codes for which the minimal weight is even.

In Chapter II we formulate and study the decoding rules for Hobbs' codes. Although the results of this chapter are related to those of Chapter I, we have preferred to give many independent proofs.

For any code A we shall denote by $w(A)$ the minimal weight of its elements, by $e(A)$ its packing integer (in our case $2e(A)+2 = w(A)$), and by $n(A)$ the length of its sequences. Further, $b(A)$ will denote the maximal length of correctable bursts: a burst is correctable if it is the only coset leader or if it is the shortest burst among the possible coset leaders. Similarly $b_c(A)$ will denote the maximal length of correctable cyclic bursts; the length of a sequence $(x_1, \ldots, x_n)$ considered as a cyclic burst being defined as follows: $\min(n-j+i)$ for all integers $i \leq j$ such that $x_i = x_{j+1} = 1$ but $x_h = 0$ if $i < h \leq j$.

Chapter I The Product of Two Codes

1. Notations and Assumptions

If A and B are two codes, we denote by AB their product, that is the code whose elements are sequences $(X_{11}, X_{12}, X_{13}, \ldots, X_{1n(B)}, X_{21}, \ldots, X_{n(A)n(B)})$ of $n(AB) = n(A)n(B)$ terms X_{ij} such that, for any fixed i, $(X_{i1}, X_{i2}, \ldots, X_{in(B)})$ is an element of B and, for any fixed j, $(X_{1j}, X_{2j}, \ldots, X_{n(A)j})$ is an element of A (see [2], [3], [4], [5]). We can clearly represent the elements of AB as matrices with $n(A)$ rows $(X_{i1}, X_{i2}, \ldots, X_{in(B)})$ and $n(B)$ columns $(X_{1j}, X_{2j}, \ldots, X_{n(A)j})$. For fixed i, the collection of the i^{th} rows of all the elements of AB form a group, hence a code, denoted B_i : it is obviously isomorphic to B. Similarly A_j will denote the code of all the j^{th} possible columns. We assume to have decoding rules for A [and for B] to correct all configurations with not more than $e(A)$ [and $e(B)$] errors, all bursts of length not more than $b(A)$ [and $b(B)$] and all cyclic bursts of length not more than $b_c(A)$ [and $b_c(B)$]. Notice that, even if $e(A) = b(A) = b_c(A) = 0$, we will always be able to detect the presence of one error, since $w(A)$ is assumed even. Similarly for B. Also, since the A_j are isomorphic to A, all our assumptions apply as well to each one of them. Similarly for the codes B_i, isomorphic to B.

In the following we shall say that an error configuration in A_j is correctable if and only if the number of errors is $\leq e(A)$, or the configuration is a burst of length $\leq b(A)$, or a cyclic burst of length $\leq b_c(A)$. Similarly for B_i.

2. The Decoding Rules for AB

The decoding proceeds in two major steps. The first consists in applying, when appropriate, the decoding rules for A to the $n(B)$ "columns" A_j and in keeping track of the work performed using a numerical function denoted ψ. The second step applies the decoding rules for

B to the $n(A)$ "rows" B_i obtained <u>after</u> the corrections required by Step I.

In precise details:

<u>Step I</u> For $j = 1, 2, \ldots, n(B)$:

α_j) no errors are detected in A_j : set $y(j) = 0$.

β_j) $\ell > 0$ errors are detected in A_j and they form a correctable configuration: correct them and set $y(j) = \ell$.

γ_j) errors are detected in A_j and they form a configuration which is not correctable: set $y(j) = -1$.

<u>Step II</u> In the matrix obtained after application of Step I:

α) no errors are detected in any B_i , $i = 1, 2, \ldots, n(A)$: accept.

β) $\ell_r > 0$ errors are detected in B_{i_r} , $r = 1, 2, \ldots, \rho$ and they form a correctable configuration for $r = 1, 2, \ldots, \alpha$ with $\alpha \leq \rho$:

1) if $\alpha = \rho$, correct in each B_{i_r} and accept;

2) if $\alpha < \rho$, let $j_1, j_2, \ldots, j_\sigma$ be all the integers j such that $y(j) = -1$ and, if $\tau = e(B) + 1 - \sigma > 0$, let $k_1, k_2, \ldots, k_\tau$ be such that $y(k_\tau) > y(j)$ for all $j \neq k_1, k_2, \ldots, k_\tau$; then if $\sigma = e(B) + 1$ [or $\sigma + \tau = e(B) + 1$], correct the errors in $B_{i_1}, B_{i_2}, \ldots, B_{i_\alpha}$ and change all the terms $x_{i_r j_1}, x_{i_r j_2}, \ldots, x_{i_r j_\sigma}$ and $[x_{i_r k_1}, x_{i_r k_2}, \ldots, x_{i_r k_\tau}]$ for $r = \alpha+1, \alpha+2, \ldots, \rho$ and accept.

γ) in all other cases, reject.

To "change" x_{ij} means to substitute it with $x_{ij} + 1$, the sum being modulo 2. It is important to emphasize that the checking equations for the rows of the received matrix need be computed only after having performed Step I. The function y plays an important role in Step II β) 2), whose formal statement is somewhat complicated. Notice that if $\sigma = e(B) + 1$, there is no need to search for the integers $k_1, k_2, \ldots$ at which y assumes its larger values; and if $\sigma < e(B) + 1$ but no integers $k_1, k_2, \ldots, k_\tau$ can be found, then we are in case γ) . The examples discussed in the next section will illuminate the situation.

3. Examples

A) Let $A = B$ be the (7,3) code generated by

1001110, 0100111, 0011101;

it is easy to check that $w(A) = 4$, $e(A) = 1$ and $b(A) = b_2(A) = 2$.

Suppose we send the zero matrix (=zero element of AB, which is a (49,9) code with $w(AB) = 16$) and we receive

$$E = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$A_1\ A_2\ A_3\ A_4\ A_5\ A_6\ A_7$

Step I has to be applied as follows:

$j = 1$: γ_1) $y(1) = -1$

$j = 2$: α_2) $y(2) = 0$

$j = 3$: β_3) gives the new third column

1
0
0
1
1
1
0

and $y(3) = 1$

$j > 3$: α_j) gives $y(j) = 0$.

We thus have to apply Step II to

$$E' = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \end{matrix}$$

Notice that Step I has increased the number of errors! We are in case β) of Step II: B_1, B_4, B_5 and B_6 have errors, which form correctable configurations only in B_1 and B_5. Thus subcase 2). But $\sigma = \tau = 1$ that is $\sigma + \tau = 2 = e(B) + 1$. Accordingly we correct errors in B_1 and B_5 and change

x_{41}, x_{61} since $y(1) = -1$

x_{43}, x_{63} since $y(3) = 1 > y(j)$ for $j \neq 3$.

This yields obviously the zero matrix. Why do we change the elements of B_4 and B_6 in the first and third columns? We know, since $y(1) = -1$, that after Step I there are still errors present in A_1; and we know now that there are errors in B_4 and B_6. Thus the intersections x_{41} and x_{61} are singled out as probably in error. But if only these two terms would be erroneous, we could have corrected them in B_4 and B_6: hence these two rows contain further errors. In all other columns we have (after Step I) an element of the code A: which to choose? By the maximum likelihood principle, the one (or those) in which we have made the greatest number of corrections (in this case, one correction). Hence we change x_{43} and x_{63}.

The next example will illustrate this idea again.

B) Using for A and B the same codes, assume we receive

$$E = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} .$$

Step I yields

$$E' = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

with $y(1)=0$, $y(2)=-1$, $y(3)=1$, $y(4)=-1$, $y(5)=0$, $y(6)=2$, and $y(7)=2$. Notice that the errors of E in A_6 and A_7 can be corrected since they are cyclic bursts of length $\leq b_c(A)$. We find now that B_3 has correctable errors, but B_2 and B_7 have errors we cannot correct. Hence again we check σ and τ: $y(2)=y(4)=-1$, that is $\sigma=2$. Since $\sigma = e(B)+1$, we correct the error of B_3 and change x_{22}, x_{24}, x_{72}, x_{74} obtaining the zero matrix. Here we disregard A_6 and A_7 because σ is "large enough": we do not need to assume that we have made erroneous changes in Step I.

C) A third example with the same codes will illustrate the correction of cyclic bursts. Suppose we receive

$$E = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

After Step I we obtain

$$E' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The $y(j)$ are easily written down, but are not important since now Step II β) 1) applies to give at once the zero matrix.

D) Assume now that A is the (4,3) code of all the sequences with even weight, and similarly B is the (5,4) code with $w(B)=2$. Then AB is the (20,12) 2-dimensional Hobbs' code. We have $e(A) = e(B) = b(A) = b(B) = b_2(A) = b_2(B) = 0$.

Suppose we receive

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix};$$

we obtain $y(1) = y(2) = y(3) = -1$, the others being $= 0$. And we detect errors in B_2. We cannot apply Step II β) 2),

because now $\sigma = 3 > e(B)+1 = 1$. Thus, we reject. And this is in proper order, since we would have had the same y's and same detection of errors in the B_c's if we had received, say

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Finally note that by our rules we would accept the matrix

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

because of Step II α). But in this connection see Theorem 2 and 3 below.

4. The Main Theorem

In order to simplify the formulation of our results it will be convenient to adopt the following notation and terminology. If E is an error configuration (that is, a matrix giving the difference between what was sent and what has been received), denote by E_j, $j = 1, 2, \ldots, n(B)$ its columns: E_j is then the configuration of the errors belonging to A_j. We shall say that E_j is A-correctable if it is correctable as error configuration in $A_j = A$ (see end of section 1). Finally, we will call a set $E_{j_1}, E_{j_2}, \ldots, E_{j_s}$ B-correctable if the sequence $(b_1, b_2, \ldots, b_{n(B)})$ with $b_{j_1} = b_{j_2} = \cdots = b_{j_s} = 1$ and $b_j = 0$ if $j \neq j_r$ is correctable as error configuration in B. This terminology can be illustrated using the examples above. Thus, in example A), E_1 and E_3 are not A-correctable, although the decoding rules for A will operate on E_3 as indicated: but E_3 is not correctable since it has more than $e(A)$ errors and is a burst longer than $b(A)$ or $b_c(A)$. In this

same example, the set E_1, E_3 is not B-correctable, for the same reasons. In example C), E_1 and E_7 are not A-correctable, but their set is B-correctable, since (1,0,0,0,0,0,1) is a cyclic burst of length $= b_c(B)$.

Theorem 1

The decoding rules given for AB correct every error configuration E satisfying any one of the following conditions:

1) The set of E_j which are not A-correctable is B-correctable;

2) The number of errors does not exceed
$$\varepsilon = [e(A)+1][e(B)+1] + e(A);$$

3) E is a burst of length not exceeding
$$\beta = \max(b(A)n(B) + b_c(B),\ e(A)n(B) + e(B) + 1);$$

4) E is a cyclic burst of length not exceeding
$$\beta_c = \max\{b_c(A)n(B) + b_c(B),\ e(A)n(B) + e(B) + 1\}.$$

Even before proving this theorem, let us remark that $w(AB) = w(A)w(B)$ and thus, under our assumption, $e(AB) = (e(A)+1)(2e(B)+2) - 1$; if $e(B) = 0$ we have then $\varepsilon = e(AB)$.

Corollary 1.1 If $e(B) = 0$, or equivalently if $w(B) = 2$, the decoding rules given above for AB correct all configurations with at most $e(AB)$ errors.

The proof of the Theorem if E satisfies 1) is straight forward. After application of Step I we will be left with errors (original or newly made) only in the non-A-correctables columns, say, $E_{j_1}, E_{j_2}, \ldots, E_{j_\rho}$: but by assumption these form a B-correctable set. Hence Step II α) or Step II β) 1) will apply. Thus all errors will be corrected. Notice in particular that if all E_j are A-correctable, then the correction of E is already performed after Step I.

Consider now case 2). If at most $e(B)$ of the E_j are not A-correctable, we are back in case 1). Thus we can assume at least $e(B)+1$ non-A-correctable E_j's. In each of these, then, there is at least

$e(A)+1$ errors: because of the value of ε, this implies also that there are at most, and thus exactly, $e(B)+1$ E_j's which are not A-correctable. Let these be $E_{j_1}, E_{j_2}, \dots, E_{j_\sigma}, E_{k_1}, E_{k_2}, \dots, E_{k_\tau}$ with $y(j_1) = y(j_2) = \dots = y(j_\sigma) = -1$ and $y(k_t) > -1$. If $\sigma = e(B)+1$, and thus $\tau = 0$, Step I followed by Step II β) will correct all errors. In fact, after Step I we will obtain a matrix E' with ones only in the columns $j_1, j_2, \dots, j_\sigma$. Since $\sigma = e(B)+1$, each row will contain at most $e(B)+1$ errors: we will thus detect them all and be in Step β) 1) or β) 2). This reasoning can be applied also if $\tau \neq 0$, provided we show $y(k_t) > y(j)$ for all $j \neq k_1, k_2, \dots, k_\tau$. We will even show $y(k_t) > m$ where m denotes the number of all the errors of E not contained in $E_{j_1}, E_{j_2}, \dots, E_{j_s}, E_{k_1}, \dots, E_{k_\tau}$. Let ℓ be the total number of errors in E, ℓ_0 the number of errors in $E_{j_1}, E_{j_2}, \dots, E_{j_\sigma}$ and ℓ_t the number of errors in E_{k_t}. Then $\ell = m + \ell_0 + \sum \ell_t \leq \varepsilon$ and $\ell_t \geq e(A)+1$. Thus, for a fixed t_0, we have

$$\ell_0 + \sum_{t \neq t_0} \ell_t \geq e(B)[e(A)+1] = \varepsilon - [2e(A)+1]$$

or

$$2e(A) + 2 - \varepsilon + \ell_0 + \sum_{t \neq t_0} \ell_t \geq 1.$$

On the other hand, $\ell_t + y(k_t) \geq w(A)$ (for any code A, the number ℓ_t of errors present plus the number $y(k_t)$ of changes made is at least $w(A)$, if the changes do not reproduce the sequence originally sent); thus

$$y(k_{t_0}) \geq w(A) - \ell_{t_0} = 2e(A) + 2 - \left[\ell - m - \ell_0 - \sum_{t \neq t_0} \ell_t\right] \geq$$

$$\geq 2e(A) + 2 - \varepsilon + m + \ell_0 + \sum_{t = t_0} \ell_t \geq m+1 > m.$$

This terminates the proof of case 2).

For case 3), remember first of all that by definition of product code, the order of transmission is "row-wise", as indicated in section 1 above. This implies that x_{ij} will be the $[(i-1)n(B) + j]^{th}$ term sent. Let E then be a burst of length at most β, starting at x_{ij}

and ending at X_{uv}:

$$[(u-1)n(B)+v]-[(i-1)n(\beta)+j]+1=(u-i)n(\beta)+v-j+1\leq\beta.$$

Thus, if $\beta=b(A)n(B)+b_c(B)$, there are at most $b_c(B)$ non-A-correctable E_j's which can be easily seen to form a B-correctable set (cyclic burst). We can thus apply case 1). If now $\beta=e(A)n(B)+e(B)+1$, there are at most $e(B)+1$ non-A-correctable E_j's: if less than $e(B)+1$, then we are again in case 1). If exactly $e(B)+1$, they will yield either $y(j)=-1$ or $y(j)\geq e(A)+1$, since each such E_j has weight at most $e(A)+1=w(A)/2$. But each of the A-correctable E_j has weight at most $e(A)$, and thus Step II β) 2) applies.
The proof of case 4) is similar.

5. Further Results

A deeper analysis of our decoding rules gives the following theorems.

Theorem 2 In the presence of $\varepsilon+1$ errors, it is possible to obtain and accept, after application of the decoding rules given above, a matrix which is not an element of AB.

In other words, the checking equations may fail to be satisfied after decoding, even if the rules specify to "accept", provided there are more than ε errors. The proof consists in giving an example of this situation, for arbitrary codes A and B.

Let $b=(b_1, b_2, \ldots, b_{n(B)})$ be an element of B with $w(b)=w(B)$ in which $b_{j_1}, b_{j_2}, \ldots, b_{j_{w(B)}}$ are the terms equal to one. Consider then an error configuration E such that E_j is a column of zeros if $j\neq j_1, j_2, \ldots, j_{e(B)+2}$ and $E_{j_1}=E_{j_2}=\cdots=E_{j_{e(B)+2}}$ is a non-A-correctable configuration of exactly $e(A)+1$ errors (such a configuration exists because of the definition of $e(A)$). E has then $[e(A)+1][e(B)+2]=\varepsilon+1$ errors. Step I will require no changes and give $y(j)=0$ or $y(j)=-1$. Step II β) 1) will apply since in each of the $e(A)+1$ rows with errors we can obtain the sequence b by changing $e(B)$ terms. The matrix so obtained, however, does not belong to AB since the columns are not

elements of A.

This situation can be easily corrected:

Theorem 3 Suppose the rules for Step II are modified to require acceptance of the decoded matrix only if

1) $y(j) > -1$ for all j, in Step II α)
2) newly computed checking equations for each A_j are satisfied, in Step II β) 1) and 2)
3) newly computed checking equations for each B_i are satisfied, in Step II β) 2).

Then no matrix will be accepted if it is not an element of AB.

The proof is obvious and will be omitted.

We can improve the correction of bursts, without modifying the decoding rules, provided in A no burst of length $b(A)+1$ [or $b_c(A)+1$] is "wrongly corrected":

Theorem 4 Assume that in A each [cyclic] burst of length $b(A)+1$ [$b_c(A)+1$] is A-correctable or belongs to a coset containing no A-correctable configurations. Then the decoding rules for AB correct any error configuration E which is a [cyclic] burst of length not exceeding $\beta = b(A)n(B) + \max\{b_c(B), e(B)+1\}$ [$\beta_c = b_c(A)n(B) + \max\{b_c(B), e(B)+1\}$].

If $\beta = b(A)n(B) + b_c(B)$ [$\beta_c = b_c(A)n(B) + b_c(B)$] this theorem is contained in Theorem 1, and there is nothing new to prove. Suppose thus $\beta = b(A)n(B) + e(B)+1$ [$\beta_c = b_c(A)n(B) + e(B)+1$]. Following the reasoning given in the proof of Theorem 1, we see that now there are at most $e(B)+1$ non-A-correctable E_j's, which are all bursts of length $b(A)+1$. Because of our assumption, we have now $y(j) = -1$ and thus Step II β) 1) or II β) 2) will apply.

Consider now the code BA. For its elements we may take those of AB, but the order of transmission is now different. Instead of sending a matrix row-wise, we send it column-wise, so that x_{ij} will now be the $[i+(j-1)n(B)]^{th}$ term. If we apply the decoding rules given above, the first two statements of Theorem 1 will still hold true, since

they are independent of the order of transmission. In particular it follows from Theorem 1, 1) that we can correct any error configuration of BA which consists of several bursts, each of length at most $b_c(A)$, provided any two consecutive bursts are separated by at least $n(A)-b(A)$ correct terms.

Chapter II

Application to Hobbs' Codes

1. Preliminaries

The decoding rules given above can be used with codes that are products of more than two factors, by convenient iteration. Moreover, the formulation of those rules can be simplified if the factors have particular properties. For instance, if $e(B)=0$ the statement of Step II β) 2) becomes much simpler. In this chapter we will explicitly give the decoding rules for δ -dimensional Hobbs' codes, which are products of δ factors. We shall use the notation of Kautz $K_1^{\delta}(n_1, n_2, \ldots, n_\delta)$ to denote the δ -dimensional Hobbs' code $(n_1 n_2 \ldots n_\delta, (n_1-1)(n_2-1)\ldots(n_\delta-1))$. Remember that (see [6])

$$w\left(K_1^{\delta}(n_1, n_2, \ldots, n_\delta)\right) = 2^{\delta};$$

$$e\left(K_1^{\delta}(n_1, n_2, \ldots, n_\delta)\right) = 2^{\delta-1}-1;$$

$$b\left(K_1^{\delta}(n_1, n_2, \ldots, n_\delta)\right) = b_c\left(K_1^{\delta}(n_1, n_2, \ldots, n_\delta)\right) =$$

$$= n_3 n_4 \ldots n_\delta + n_4 n_5 \ldots n_\delta + \cdots + n_\delta + 1.$$

It is also well known that $K_1^{\delta}(n_1, n_2, \ldots, n_\delta)$ is the product $K_1'(n_1)\, K_1'(n_2)\ldots K_1'(n_\delta)$, with the factors taken in this order. An

element of K_1^δ is thus a sequence of the form

$$(x_{11\cdots11},\ x_{11\cdots12},\ \cdots\ \cdots\ x_{11\cdots1n_\delta},$$
$$x_{11\cdots21},\ x_{11\cdots22},\ \cdots\ \cdots\ x_{11\cdots2n_\delta},$$
$$\cdots\ \cdots\ \cdots\ \cdots\ \cdots$$
$$\cdots\ \cdots\ \cdots\ \cdots\ \cdots$$
$$x_{n_1n_2\cdots n_{\delta-1}1},\ x_{n_1n_2\cdots n_{\delta-1}2},\ \cdots\ \cdots\ x_{n_1n_2\cdots n_{\delta-1}n_\delta}),$$

in which the k^{th} term has indices $i_1, i_2, \cdots, i_\delta$ satisfying

$$k = i_\delta + (i_{\delta-1}-1)n_\delta + (i_{\delta-2}-1)n_\delta n_{\delta-1} + \cdots + (i_1-1)n_\delta n_{\delta-1}\cdots n_2.$$

For some ρ with $1 \le \rho < \delta$ consider now the set of those sub-sequences, for which $i_{\rho+1} = \alpha_{\rho+1}, \cdots, i_\delta = \alpha_\delta$ are given values. This set is a subcode of dimension ρ for which we shall use the notation $K_1^\rho(n_1, \cdots, n_\rho;\ \alpha_{\rho+1}, \cdots, \alpha_\delta)$; it is obviously isomorphic to $K_1^\rho(n_1, \cdots, n_\rho)$. We can clearly obtain $n_{\rho+1} n_{\rho+2} \cdots n_\delta$ different such subcodes of dimension ρ.

This structure of K_1^δ suggests that theorems for this kind of code are best established by using induction proofs. That is why we consider now also $(\rho+1)$-dimensional subcodes

$$K_1^{\rho+1}(n_1, \cdots, n_{\rho+1};\ \alpha_{\rho+2}, \cdots, \alpha_\delta) \quad \text{isomorphic to } K_1^{\rho+1}(n_1, \cdots, n_{\rho+1}) =$$
$$= K_1^\rho(n_1, \cdots, n_\rho) \cdot K_1^1(n_{\rho+1}).$$

Up to isomorphisms, then we can write

$$K_1^{\rho+1}(n_1, \cdots, n_{\rho+1};\ \alpha_{\rho+2}, \cdots, \alpha_\delta) = K_1^\rho(n_1, \cdots, n_\rho;\ i_{\rho+1};\ \alpha_{\rho+2}, \cdots, \alpha_\delta) \cdot$$
$$\cdot K_1^1(i_1, \cdots, i_\rho;\ n_{\rho+1};\ \alpha_{\rho+2}, \cdots, \alpha_\delta),$$

where the i_ν are fixed integers. This decomposition will be used in applying the decoding rules of Chapter I to obtain a decoding procedure for K_1^δ.

2. The Decoding Rules for K_1^δ.

We will have δ major steps: step number $\rho+1$ applies to all

one-dimensional subcodes of type $K_1'(i_1, \ldots, i_\rho ; \eta_{\rho+1} ; \alpha_{\rho+2}, \ldots, \alpha_\delta)$. We will again have a counting function ψ which associates to each $K_1^\rho(\eta_1, \ldots, \eta_\rho ; \alpha_{\rho+1}, \ldots, \alpha_\delta)$ an integer $\psi(\alpha_{\rho+1}, \ldots, \alpha_\delta)$. (Observe that the parameters $\rho, \alpha_{\rho+1}, \ldots, \alpha_\delta$ uniquely determine the subcode $K_1^\rho(\eta_1, \ldots, \eta_\rho ; \alpha_{\rho+1}, \ldots, \alpha_\delta)$.)

To compute $\psi(\alpha_{\rho+1}, \ldots, \alpha_\delta)$ in the steps below will mean: set $\psi(\alpha_{\rho+1}, \ldots, \alpha_\delta) = -1$ if there are integers i, j, $i \neq j$, such that $\psi(i, \alpha_{\rho+1}, \ldots, \alpha_\delta) = \psi(j, \alpha_{\rho+1}, \ldots, \alpha_\delta) = -1$; otherwise set $\psi(\alpha_{\rho+1}, \ldots, \alpha_\delta)$ equal to the number of terms $x_{i_1 \ldots i_\rho \alpha_{\rho+1} \ldots \alpha_\delta}$ of the originally received sequence which differ from the corresponding terms of the sequence obtained after application of the first ρ steps.

To compute the checking equation in $K_1'(i_1, \ldots, i_\rho ; \eta_{\rho+1} ; \alpha_{\rho+2}, \ldots, \alpha_\delta)$ will mean to compute, modulo 2, $\sum_{i=0}^{\eta_{\rho+1}-1} x_{i_1 \ldots i_\rho\, i\, \alpha_{\rho+2} \ldots \alpha_\delta}$; if this adds to one, we detect errors.

Step I Compute the checking equation in $K_1'(\eta_1 ; \alpha_2, \ldots, \alpha_\delta)$ for all $\alpha_\delta = 1, 2, \ldots, \eta_\sigma$, $\sigma = 2, \ldots, \delta$:

α) No errors are detected in $K_1'(\eta_1 ; \alpha_2, \ldots, \alpha_\delta)$: set $\psi(\alpha_2, \ldots, \alpha_\delta) = 0$.

β) errors are detected in $K_1'(\eta_1 ; \alpha_2, \ldots, \alpha_\delta)$: set $\psi(\alpha_2, \ldots, \alpha_\delta) = -1$.

It is easy to recognize that this is Step I of the previous chapter, incorporating the assumption $A_j = K_1'(\eta_1 ; \alpha_2, \ldots, \alpha_\delta)$.

Having applied step ρ, we apply:

Step $\rho+1$. Compute the checking equation in each $K_1'(i_1, \ldots, i_\rho ; \eta_{\rho+1} ; \alpha_{\rho+2}, \ldots, \alpha_\delta)$, for all $i_\tau = 1, 2, \ldots, \eta_\tau$, $\tau = 1, \ldots, \rho$ and all $\alpha_\sigma = 1, 2, \ldots, \eta_\sigma$, $\sigma = \rho+2, \ldots, \delta$:

α) No errors are detected in any $K_1'(i_1, \ldots, i_\rho ; \eta_{\rho+1} ; \alpha_{\rho+2}, \ldots, \alpha_\delta)$: compute $\psi(\alpha_{\rho+2}, \ldots, \alpha_\delta)$.

β) errors are detected in $K_1'(i_{1r}, \ldots, i_{\rho r} ; \eta_{\rho+1} ; \alpha_{\rho+2}, \ldots, \alpha_\delta)$ for $r = 1, \ldots, \ell$, and there exists an integer j such that, for all $i_{\rho+1} \neq j$

$$\psi(j, \alpha_{\rho+2}, \ldots, \alpha_\delta) > \psi(i_{\rho+1}, \alpha_{\rho+2}, \ldots, \alpha_\delta) \geq 0$$

or

$$\psi(i_{\rho+1}, \alpha_{\rho+2}, \ldots, \alpha_\delta) > \psi(j, \alpha_{\rho+2}, \ldots, \alpha_\delta) = -1 :$$

Change $x_{i_{1r} \ldots i_{\rho r}\, j\, \alpha_{\rho+2} \ldots \alpha_\delta}$ for $r = 1, \ldots, \ell$ and compute $\psi(\alpha_{\rho+2}, \ldots, \alpha_\delta)$.

γ) in all other cases: set $y_i(\alpha_{s+2}, \ldots, \alpha_\delta) = -1$.

We can recognize here Step II of Chapter I, formulated for $A_j = K_1^{\rho}(n_1, \ldots, n_{s+1}; \alpha_{s+2}, \ldots, \alpha_\delta)$ and $B = K_1'$. In particular, the product structure of K_1^{s+1} and the relation $c(B) = 0$ have been utilized.

The final step is very similar to Step II of Chapter I.

Step δ Compute the checking equation in $K_1'(i_1, \ldots, i_{\delta-1}; n_\delta)$ for all $i_r = 1, 2, \ldots, n_r$, $r = 1, 2, \ldots, \delta-1$.

α) No errors are detected in any $K_1'(i_1, \ldots, i_{\delta-1}; n_\delta)$ and $y_i(i_\delta) \geq 0$ for all i_δ : accept.

β) errors are detected in $K_1'(i_{1r}, \ldots, i_{(\delta-1)r}; n_\delta)$ for $r = 1, \ldots, \ell$ and there exists an integer j such that for all $i_\delta \neq j$ $y_i(j) > y_i(i_\delta) \geq 0$ or $y_i(i_\delta) > y_i(j) = -1$:

Change $x_{i_{1r} \ldots i_{(\delta-1)r} j}$ for $r = 1, \ldots, \ell$ and accept.

γ) in all other cases: reject.

Before discussing the performance of these decoding rules, we give an example in which the corrections in all the different steps are carried out in detail.

3. An example

Consider $K_1^4(3,3,3,4)$: it is a (108,24) code with $w = 16$, $e = 7$, $b = b_e = 17$. An element of this code is a sequence $\{x_{i_1 i_2 i_3 i_4}\}$ with

$i_1 = 1,2,3$

$i_2 = 1,2,3$

$i_3 = 1,2,3$

$i_4 = 1,2,3,4$.

Assume we receive a sequence in which the non-zero terms are

x_{1111}	(1)
x_{1113}	(3)
x_{1124}	(8)
x_{1132}	(10)
x_{3322}	(102)
x_{3324}	(104)
x_{3331}	(105)
x_{3333}	(107)

The numbers in parentheses give the position in the sequence.

Since not two of the x's belong to the same $K_1'(n_1; \alpha_2, \alpha_3, \alpha_4)$ (no two have the same last three indices), Step I gives $y(\alpha_2, \alpha_3, \alpha_4) = 0$ for all values of the α's, but for:

$y(1,1,1) = -1$ $\quad$ $y(3,2,2) = -1$
$y(1,1,3) = -1$ $\quad$ $y(3,2,4) = -1$
$y(1,2,4) = -1$ $\quad$ $y(3,3,1) = -1$
$y(1,3,2) = -1$ $\quad$ $y(3,3,3) = -1$

In Step II, for $\alpha_3 = 1$, $\alpha_4 = 1$, we detect an error in $K_1'(1; 3; 1, 1)$, namely x_{1111}; moreover, we find $y(1,1,1) = -1 \neq y(i_2, 1, 1) = 0$ for $i_2 \neq 1$. Thus, by II β), we change x_{1111} and compute $y(1,1) = 1$. For $\alpha_3 = 1$, $\alpha_4 = 2$, we detect no errors in any $K_1'(i_1; 3; 1, 2)$; and since $y(i_2, 1, 2) \neq -1$, we set $y(1,2) = 0$. For $\alpha_3 = 1$, $\alpha_4 = 4$; $\alpha_3 = 2$, $\alpha_4 = 1$; $\alpha_3 = 2$, $\alpha_4 = 3$ and $\alpha_3 = 3$, $\alpha_4 = 4$ the situation is analogous, yielding zero for the corresponding values of y.

The cases $\alpha_3 = 1$, $\alpha_4 = 3$; $\alpha_3 = 2$, $\alpha_4 = 2$; $\alpha_3 = 3$, $\alpha_4 = 1$; $\alpha_3 = 3$, $\alpha_4 = 2$, and $\alpha_3 = 3$, $\alpha_4 = 3$ are similar to the first case considered ($\alpha_3 = \alpha_4 = 1$). They all yield 1 for the corresponding value of y, each requiring the correction of, respectively,

$$x_{1113};\ x_{3322};\ x_{3331};\ x_{1132};\ x_{3333}.$$

We have only to consider now, always for Step II, the case $\alpha_3 = 2$ and $\alpha_4 = 4$. Here we detect errors in $K_1'(1;3;2,4)$ as well as in $K_1'(3;3;2,4)$. But we cannot apply II β) since we have $y(1,2,4) = -1$ as well as $y(3,2,4) = -1$. Thus we are in II γ) and set $y(2,4) = -1$.

After this, we have a sequence with only two non-zero terms, namely x_{1124} and x_{3324}. To this sequence we now apply Step III.

If $\alpha_4 \neq 4$ we do not detect errors in any $K_1'(i_1, i_2; 3; \alpha_4)$; moreover, $y(\alpha_3, \alpha_4) \neq -1$ if $\alpha_4 \neq 4$, thus we compute y by determining the number of terms $x_{i_1 i_2 i_3 \alpha_4}$ which now differ from the original ones:

$y(1) = 2$ $\quad$ $y(2) = 2$ $\quad$ $y(3) = 2$.

If $\alpha_4 = 4$, we detect errors in $K_1'(1,1;3;4)$ and $K_1'(3,3;3;4)$. Also $y(2,4) = -1 < y(i_3,4)$ for $i_3 \neq 2$. Thus we change $x_{1,1,2,4}$ and $x_{3,3,2,4}$ and set $y(4) = 2$.

We have now the sequence of all zeros [and $y(\alpha_k) > -1$ for all α_k]; thus Step IV α) applies and we accept.

4. First Results

The following notation will be useful. If E is an error configuration (that is, a sequence giving the difference between what was sent and what has been received), and if $\rho < \delta$, we denote by $E(\alpha_{\rho+1}, \ldots, \alpha_\delta)$ that subconfiguration which belongs to $K_1^\rho(n_1, \ldots, n_\rho; \alpha_{\rho+1}, \ldots, \alpha_\delta)$; if $\rho = \delta$, we shall set $E(\alpha_{\rho+1}, \ldots, \alpha_\delta) = E$. By E^ρ we shall represent the error configuration present after step number ρ (that is, that sequence which gives the difference between what was sent and what is present after step number ρ); similarly for $E^\rho(\alpha_{\rho+1}, \ldots, \alpha_\delta)$. If by ϕ we denote the configuration of no errors (the zero sequence), the decoding will be successful when $E^\delta = \phi$. If, finally, X_ρ denotes an element of any K_1^ρ , the decoding will be successful or leave undetected errors if $E^\delta = X_\rho$.

The following lemmas will help prove our theorems below and, at the same time, will clarify the operation of the decoding rules.

Lemma 1 For any error configuration E and $1 \leq \rho \leq \delta$, the following two propositions are equivalent:

1. $E^\rho(\alpha_{\rho+1}, \ldots, \alpha_\delta) = X_\rho$

2. $y(\alpha_{\rho+1}, \ldots, \alpha_\delta) \geq 0$ if $\rho < \delta$; step δ requires acceptance, if $\rho = \delta$.

Notice that step δ requires acceptance exactly under those circumstances under which any previous step yields $y \geq 0$. Thus we may omit the distinction between $\rho < \delta$ and $\rho = \delta$. The lemma is trivially true for $\rho = 1$. Let us then assume it for ρ and prove it for $\rho + 1$.

Suppose first that $E^{s+1}(\alpha_{s+2}, \dots, \alpha_s) = X_{s+1}$. If the $(s+1)^{st}$ step did not require any changes, then already $E^s(i_{s+1}, \alpha_{s+2}, \dots, \alpha_s) = X_s$ for all $i_{s+1} = 1, 2, \dots, n_{s+1}$; thus by induction assumption $\mu(i_{s+1}, \alpha_{s+2}, \dots, \alpha_s) \geq 0$ and by the decoding rules $\mu(\alpha_{s+2}, \dots, \alpha_s) \geq 0$. If we had some changes in the $(s+1)^{st}$ step, by definition $\mu(\alpha_{s+2}, \dots, \alpha_s) \geq 0$.

The proof of the converse is not as simple. We first assume to have obtained $\mu(\alpha_{s+2}, \dots, \alpha_s) \geq 0$ by applying case $\alpha)$ of step $s+1$. If $\mu(i_{s+1}, \alpha_{s+2}, \dots, \alpha_s) \geq 0$ for all i_{s+1}, then by induction assumption $E^s(i_{s+1}, \dots, \alpha_s) = X_s$; and since no errors are detected in any $K_1'(i_1, \dots, i_s ; n_{s+1} ; \alpha_{s+2}, \dots, \alpha_s)$, clearly $E^{s+1}(\alpha_{s+2}, \dots, \alpha_s) = X_{s+1}$. If $\mu(i_{s+1}, \alpha_{s+2}, \dots, \alpha_s) = -1$ for at least two values of i_{s+1}, then by definition $\mu(\alpha_{s+2}, \dots, \alpha_s) = -1$, contrary to the assumption. Thus, the only case to consider is $\mu(j, \alpha_{s+2}, \dots, \alpha_s) = -1$, $\mu(i_{s+1}, \alpha_{s+2}, \dots, \alpha_s) \geq 0$ if $i_{s+1} \neq j$. By induction assumption $E^s(i_{s+1}, \alpha_{s+2}, \dots, \alpha_s) = X_s$ if $i_{s+1} \neq j$ and $E^s(j, \alpha_{s+2}, \dots, \alpha_s) \neq X_s$. Moreover $E^s(i_{s+1}, \alpha_{s+2}, \dots, \alpha_s) \neq \phi$ for some i_{s+1} , say $i_{s+1} = h_1, h_2, \dots, h_r$: otherwise we could not be in case $\alpha)$ of step $s+1$. Let $X_{s,i} = E^s(h_i, \alpha_{s+2}, \dots, \alpha_s)$ and denote by $X'_{s,i}$ the sequence obtained from $X_{s,i}$ by changing the $(s+1)^{st}$ subscript from h_i to j : $X'_{s,i}$ is thus an element of the code $K_1^s(n_1, \dots, n_s ; j ; \alpha_{s+2}, \dots, \alpha_s)$.

Set

$$\bar{E}^s(j, \alpha_{s+2}, \dots, \alpha_s) = E^s(j, \alpha_{s+2}, \dots, \alpha_s) + \sum_{i=1}^{r} X'_{s,i}$$

$$\bar{E}^s(h_i, \alpha_{s+2}, \dots, \alpha_s) = E^s(h_i, \alpha_{s+2}, \dots, \alpha_s) + X_{s,i} = \phi$$

$$\bar{E}^s(i_{s+1}, \alpha_{s+2}, \dots, \alpha_s) = E^s(i_{s+1}, \alpha_{s+2}, \dots, \alpha_s) = \phi \quad i_{s+1} \neq j, h_i .$$

The union of these $\bar{E}^s$ is an error configuration $\bar{E}^s(\alpha_{s+2}, \dots, \alpha_s)$ which differs from $E^s(\alpha_{s+2}, \dots, \alpha_s)$ only by the addition of the $X'_{s,i} \cup X_{s,i}$, which are elements of $K_1^{s+1}(n_1, \dots, n_{s+1} ; \alpha_{s+2}, \dots, \alpha_s)$. The two error configurations are thus not distinguishable, and case $\alpha)$ of step $s+1$ still applies with $\mu(\alpha_{s+2}, \dots, \alpha_s) \geq 0$. But notice that in each $K_1'(i_1, \dots, i_s ; n_{s+1} ; \alpha_{s+2}, \dots, \alpha_s)$ we have exactly one term in common with $K_1^s(n_1, \dots, n_s ; j ; \alpha_{s+1}, \dots, \alpha_s)$. Since this subcode and only this

subcode among the $K_1^{\rho}(n_1, \ldots, n_\rho; i_{\rho+1}; \alpha_{\rho+2}, \ldots, \alpha_\delta)$ contains errors, we reach a contradiction. Hence $y_\rho(j, \alpha_{\rho+2}, \ldots, \alpha_\delta) = -1$ is not possible.

Assume now to obtain $y_\rho(\alpha_{\rho+2}, \ldots, \alpha_\delta) \geq 0$ by applying case $\beta)$ of step $\rho+1$. The two alternatives of this case can be now formulated: there is a j such that $E^{\rho+1}(i_{\rho+1}, \alpha_{\rho+2}, \ldots, \alpha_\delta) = E^{\rho}(i_{\rho+1}, \alpha_{\rho+2}, \ldots, \alpha_\delta) = X_\rho$ for $i_{\rho+1} \neq j$ and $E^{\rho+1}(j, \alpha_{\rho+2}, \ldots, \alpha_\delta) \neq E^{\rho}(j, \alpha_{\rho+2}, \ldots, \alpha_\delta)$ with the two possibilities $E^{\rho}(j, \alpha_{\rho+2}, \ldots, \alpha_\delta) = X_\rho$ or $\neq X_\rho$. In either alternative we can proceed as above in case $\alpha)$ to show that necessarily the union of all the $E^{\rho+1}(i_{\rho+1}, \alpha_{\rho+2}, \ldots, \alpha_\delta)$, that is, $E^{\rho+1}(\alpha_{\rho+2}, \ldots, \alpha_\delta)$, form an element $X_{\rho+1}$.

We shall say that $y_\rho(\alpha_{\rho+1}, \ldots, \alpha_\delta)$ gives a true value if either $y_\rho(\alpha_{\rho+1}, \ldots, \alpha_\delta) = -1$ or $E^{\rho}(\alpha_{\rho+1}, \ldots, \alpha_\delta) = \phi$.
We have then:

Lemma 2 If, $y_\rho(i_{\rho+1}, \alpha_{\rho+2}, \ldots, \alpha_\delta)$ gives a true value for all $i_{\rho+1} = 1, 2, \ldots, n_{\rho+1}$, then so does also $y_\rho(\alpha_{\rho+2}, \ldots, \alpha_\delta)$.

If $y_\rho(\alpha_{\rho+2}, \ldots, \alpha_\delta) = -1$, there is nothing to prove. So assume $y_\rho(\alpha_{\rho+2}, \ldots, \alpha_\delta) \geq 0$. We have two cases. Either all $y_\rho(i_{\rho+1}, \alpha_{\rho+2}, \ldots, \alpha_\delta) \geq 0$ or exactly one is equal to -1. In the first case, by assumption, $E^{\rho}(i_{\rho+1}, \ldots, \alpha_\delta) = \phi$ and thus also $E^{\rho}(\alpha_{\rho+2}, \ldots, \alpha_\delta) = \phi$ and a fortiori $E^{\rho+1}(\alpha_{\rho+2}, \ldots, \alpha_\delta) = \phi$. In the second case, case $\beta)$ of step $\rho+1$ applies to correct all errors present in $E^{\rho}(\alpha_{\rho+2}, \ldots, \alpha_\delta)$.

Corollary If, for some ρ, $1 \leq \rho < \delta$, all $y_\rho(\alpha_{\rho+1}, \ldots, \alpha_\delta)$ give true values, and if step δ requires acceptance, then $E^{\delta} = \phi$.

In fact, again, step δ requires acceptance in the circumstances under which previous steps would yield $y \geq 0$.

Lemma 3 If $E(\alpha_{\rho+1}, \ldots, \alpha_\delta)$ contains $\ell \leq 2^{\rho-1}$ errors, $y_\rho(\alpha_{\rho+1}, \ldots, \alpha_r)$ gives a true value. Moreover, if $\ell < 2^{\rho-1}$, then $y_\rho(\alpha_{\rho+1}, \ldots, \alpha_r) \geq 0$.

If $\rho = 1$, the lemma is obvious. So assume it true for ρ, and prove it for $\rho+1$. Let then $E(\alpha_{\rho+2}, \ldots, \alpha_\delta)$ contain at most 2^{ρ} errors, but assume that $y_\rho(\alpha_{\rho+2}, \ldots, \alpha_\delta)$ does not give a true value. This implies $y_\rho(\alpha_{\rho+2}, \ldots, \alpha_r) \geq 0$ and $E^{\rho+1}(\alpha_{\rho+2}, \ldots, \alpha_\delta) \neq \phi$. By

Lemma 2, there is j such that $y(j, \alpha_{s+2}, \ldots, \alpha_\delta)$ does not give a true value, that is $y(j, \alpha_{s+2}, \ldots, \alpha_\delta) \geq 0$ and $E^s(j, \alpha_{s+2}, \ldots, \alpha_\delta) \neq \phi$. Then, by induction assumption, $E(j, \alpha_{s+2}, \ldots, \alpha_\delta)$ has strictly more than 2^{s-1} errors and consequently each $E(i_{s+1}, \alpha_{s+2}, \ldots, \alpha_\delta)$ has less than 2^{s-1} errors, for $i_{s+1} \neq j$. Thus either $E^s(i_{s+1}, \alpha_{s+2}, \ldots, \alpha_\delta) = \phi$ or $y(i_{s+1}, \alpha_{s+2}, \ldots, \alpha_\delta) = -1$. If $E^s(i_{s+1}, \alpha_{s+2}, \ldots, \alpha_\delta) = \phi$ for all $i_{s+1} \neq j$, then step $s+1$ corrects all the errors of $E^s(j, \alpha_{s+2}, \ldots, \alpha_\delta)$ giving $E^{s+1}(\alpha_{s+2}, \ldots, \alpha_\delta) = \phi$, or $y(\alpha_{s+2}, \ldots, \alpha_\delta) = -1$: both alternatives are contrary to the assumptions: If $y(i_{s+1}, \alpha_{s+2}, \ldots, \alpha_\delta) = -1$, then only for one value, say h, of i_{s+1}, since $y(\alpha_{s+2}, \ldots, \alpha_\delta) \geq 0$. By induction assumption, then, $E(h, \alpha_{s+2}, \ldots, \alpha_\delta)$ has exactly 2^{s-1} errors; $E(\alpha_{s+2}, \ldots, \alpha_\delta)$ contains then these 2^{s-1} errors as well as the more than 2^{s-1} errors of $E(j, \alpha_{s+2}, \ldots, \alpha_\delta)$: a contradiction again.

We have now only to prove the second statement of the lemma. Assume $y(\alpha_{s+2}, \ldots, \alpha_\delta) = -1$. If this is so because $y(i_{s+1}, \alpha_{s+2}, \ldots, \alpha_\delta) = -1$ for at least two values of i_{s+1}, we can apply our induction assumption and immediately conclude that $E(\alpha_{s+2}, \ldots, \alpha_\delta)$ has at least 2^s errors. Thus $y(i_{s+1}, \alpha_{s+2}, \ldots, \alpha_\delta) \geq 0$ for all i_{s+1}, but we cannot apply step $s+1, \beta)$. This implies that $E^s(i_{s+1}, \alpha_{s+2}, \ldots, \alpha_\delta) \neq 0$ for at least two values of i_{s+1} : by Lemma 1, these subconfigurations are elements X_s, thus contain each at least 2^s errors. $E(\alpha_{s+2}, \ldots, \alpha_\delta)$ has thus at least 2^{s+1} errors.

Corollary If $E(\alpha_{s+1}, \ldots, \alpha_\delta)$ contains less than 2^{s-1} errors, then $E^s(\alpha_{s+1}, \ldots, \alpha_\delta) = \phi$.

5. The Performance

We can now prove our main result for Hobbs' codes.

Theorem 5 The decoding rules for $K_1^\delta(n_1, n_2, \ldots, n_\delta)$ accept only elements of $K_1^\delta(n_1, \ldots, n_\delta)$ and correct every error configuration E for which there exists an integer s, $1 \leq s \leq \delta$ such that one of the following conditions is satisfied:

1) Each $E(\alpha_{s+1}, \ldots, \alpha_\delta)$ contains at most $2^{s-1}-1$ errors $= e(K_1^s)$

2) $\rho < \delta$ and no $E(\alpha_{\rho+1}, \dots, \alpha_\delta)$ contains more than $2^{\rho-1}$ errors; and whenever $E(\alpha_{\rho+1}, \alpha_{\rho+2}, \dots, \alpha_\delta)$ contains $2^{\rho-1}$ errors, then every $E(j_{\rho+1}, \alpha_{\rho+2}, \dots, \alpha_\delta)$ with $j_{\rho+1} \neq \alpha_{\rho+1}$ contains less than $2^{\rho-1}$ errors.

3) $\rho > 2$ and each $E(\alpha_{\rho+1}, \dots, \alpha_\delta)$ is a burst or a cyclic burst of length at most $b(K_1^\rho) = n_3 n_4 \dots n_\rho + n_4 \dots n_\rho + \dots + n_\rho + 1$.

Moreover, in case 1) and 3) all the errors will be corrected after step ρ; while in case 2), they will be corrected after step $\rho+1$.

An immediate consequence is:

Corollary The decoding rules for K_1^δ correct up to the theoretical limits $\epsilon(K_1^\delta)$ and $b(K_1^\delta) = b_c(K_1^\delta)$.

The first statement of the theorem is contained in Lemma 1. To prove 1), we apply the Corollary to Lemma 3 to obtain $E^\rho(\alpha_{\rho+1}, \dots, \alpha_\delta) = \phi$ for all $\alpha_{\rho+1}, \dots, \alpha_\delta$; and thus $E^\rho = \phi$ and $E^\delta = \phi$.

Consider now 2). By Lemma 3 and its corollary, $E^\rho(j_{\rho+1}, \alpha_{\rho+2}, \dots, \alpha_\delta) = \phi$ if $j_{\rho+1} \neq \alpha_{\rho+1}$. Thus step $\rho+1$ will correct all the errors (if any) of $E^\rho(\alpha_{\rho+1}, \dots, \alpha_\delta)$ giving $E^{\rho+1}(\alpha_{\rho+2}, \dots, \alpha_\delta) = \phi$ for all $\alpha_{\rho+2}, \dots, \alpha_\delta$.

Let us now prove 3), under the assumption $\rho = 3$. If each $E(\alpha_4, \dots, \alpha_\delta)$ is a burst or a cyclic burst of length at most n_3+1, every $E(i_3, \alpha_4, \dots, \alpha_\delta)$ contains at most $\alpha = 2^{2-1}$ errors; and at most one of them, say $E(j, \alpha_4, \dots, \alpha_\delta)$ will have exactly 2 errors. We can thus apply the second part of the theorem. To prove 3) in general, we use once more induction on ρ. Setting $K_1^{\rho+1} = K_1^\rho K_1^1$, by Theorem 4 we can correct in $K_1^{\rho+1}$ ($E^{\rho+1} = \phi$) any burst (cyclic or not) of length at most $b(K_1^\rho) n_{\rho+1} + 1$ where, by induction assumption, $b(K_1^\rho) = n_3 n_4 \dots n_\rho + n_4 n_5 \dots n_\rho + \dots + n_\rho + 1$.

In Theorem 5, which is now completely proven, we decompose K_1^δ into disjoint subcodes, all of the same dimension ρ. This decomposition induces the decomposition of E into the $E(\alpha_{\rho+1}, \dots, \alpha_\delta)$ of the theorem. Clearly, we can decompose K_1^δ also in disjoint subcodes of dimensions $\rho_1, \rho_2, \dots, \rho_\nu, \dots$. If the corresponding subconfigurations are denoted $E_1, E_2, \dots, E_\nu, \dots$ let us assume that the decoding rules, when applied to $K_1^{\rho_\nu}$, yield $E_\nu^{\rho_\nu} = \phi$, for each ν. The disjointness of the subcodes insures us then that, when applying the decoding rules to K_1^δ,

we will also obtain $E_\nu^{S_\nu} = \phi$ for all ν ; and hence also $E^S = \phi$. We have thus established the following result, of which Theorem 5 can be considered as a particular, more explicit case.

Theorem 6 Let E be an error configuration in $K_1^S(n_1, \ldots, n_S)$ for which there exists a decomposition of K_1^S in pair-wise disjoint subcodes $K_1^{S_\nu}$, $\nu = 1, 2, \ldots$ with the following property: for each ν , the decoding rules for $K_1^{S_\nu}$ correct the errors of E belonging to it. Then the decoding rules for K_1^S correct E.

For an example let $S = 5$ and consider $K_1^5(n_1, n_2, n_3, n_4, n_5)$ with (arbitrary n_1, n_2) and $n_3 = 5$, $n_4 = 4$, $n_5 = 3$. For error configuration E we assume:

at most $8 = 2^{4-1}$ errors with 5th index equal to 1 , that is in $E_1 = E(1)$

at most $4 = 2^{3-1}$ errors with 4th index equal to 1 and 5th index equal to 2 , that is in $E_2 = E(1,2)$

a burst of length at most $6 = n_3 + 1$ in $E_3 = E(2,2)$

at most $2 = 2^{2-1}$ error in each $E_{3+\alpha_3} = E(\alpha_3, 3, 2)$

at most $4 = 2^{3-1}$ errors in $E_9 = E(4,2)$

a burst of length at most $25 = n_3 n_4 + n_4 + 1$ in $E_{10} = E(3)$.

In this description we have already decomposed E and K_1^5 as required by the theorem. The subcodes are as follows:

$$K_1^{S_1} = K_1^4(n_1, n_2, n_3, n_4 ; 1)$$

$$K_1^{S_2} = K_1^3(n_1, n_2, n_3 ; 1, 2)$$

$$K_1^{S_3} = K_1^3(n_1, n_2, n_3 ; 2, 2)$$

$$K_1^{S_{3+\alpha_3}} = K_1^2(n_1, n_2 ; \alpha_3, 3, 2)$$

$$K_1^{S_9} = K_1^3(n_1, n_2, n_3 ; 4, 2)$$

$$K_1^{S_{10}} = K_1^4(n_1, n_2, n_3, n_4 ; 3).$$

If we apply the decoding rules to K_1^5 we will correct $E_4, E_5, \dots, E_8$ after the second step ($\alpha = \beta_4 = \dots = \beta_8$) : in symbols $E^2_{3+\alpha_3} = \phi$. Similarly $E^3_2 = E^3_3 = E^3_9 = \phi$ and $E^4_1 = E^4_{10} = \phi$.

6. Additional Remarks

For short, let us call decoding I the one described in Chapter I, and decoding II the one described in Chapter II. Then we can say that decoding II is essentially obtained by iterating decoding I thanks to the relation

$$K_1^\delta(n_1, n_2, \dots, n_\delta) = \left(\dots\left(\left(K_1'(n_1)\,K_1'(n_2)\right)K_1'(n_3)\right)\dots\right)K_1'(n_\delta).$$

Since the two decodings agree if $\delta = 2$, we can also express their relation as follows. Decoding II for $K_1^\delta(n_1, n_2, \dots, n_\delta)$ is obtained by applying decoding II to $K_1^{\delta-1}(n_1, \dots, n_{\delta-1})$ (and to $K_1'(n_\delta)$!) and then decoding I to $K_1^{\delta-1}(n_1, \dots, n_{\delta-1})\,K_1'(n_\delta)$.

Suppose now that we apply decoding II to $K_1^\rho(n_1, \dots, n_\rho)$ and to $K_1^{\delta-\rho}(n_{\rho+1}, \dots, n_\delta)$, and then decoding I to their product $K_1^\rho(n_1, \dots, n_\rho)\,K_1^{\delta-\rho}(n_{\rho+1}, \dots, n_\delta)$, for some $\rho < \delta-1$. Then Theorem 1 and Theorem 5 insure us of correcting only up to ε errors, where

$$\varepsilon = 2^{\rho-1}2^{\delta-\rho-1} + 2^{\rho-1} - 1 = 2^{\delta-2} + 2^{\rho-1} - 1 < 2^{\delta-1} - 1;$$

and similarly for the burst length β.

The remark at the end of Chapter I however suggests to apply decoding II to $K_1^\rho(n_1, \dots, n_\rho)$ and to $K_1^{\delta-\rho}(n_{\rho+1}, \dots, n_\delta)$ and then decoding I to the inverted product $K_1^{\delta-\rho}(n_{\rho+1}, \dots, n_\delta)\,K_1^\rho(n_1, \dots, n_\rho)$. This is equivalent to applying the decoding discussed above, but to send the sequences of $x_{i_1 i_2 \dots i_\delta}$, not in the order described in section 1, but in increasing order of K, where

$$K = i_\rho + (i_{\rho-1} - 1)n_\rho + (i_{\rho-2} - 1)n_\rho n_{\rho-1} + \dots + (i_1 - 1)n_\rho n_{\rho-1} \dots n_2 + \\ + \left\{(i_\delta - 1) + (i_{\delta-1} - 1)n_\delta + \dots + (i_{\rho+1} - 1)n_\delta n_{\delta-1} \dots n_{\rho+2}\right\} n_\rho n_{\rho-1} \dots n_1.$$

Under these assumptions we can correct any series of bursts provided that each one has a length not exceeding $b(K, f)$ and that any two are separated by at least $n_1 n_2 \ldots n_f - b_f$ correct terms. At the same time, an additional number of errors not exceeding $2^{\delta-g-1}-1$ will also be corrected.

References

1. P. Calingaert, Two-Dimensional Parity Checking, Jour. ACM 8(1961) p. 186-200

2. L. Calabi, Additions and Multiplications of Codes, Tech. Memo. 11 - 3471, PML, June 1959

3. D. Slepian, Some Further Theory of Group Codes, B.S.T. Jour. 39 (1960) p. 1219-1252

4. V.W. Peterson, Error-Correcting Codes, M.I.T. Press and J. Wiley + Sons, New York, 1961

5. L. Calabi and R. Darst, A Study of the Sum and Product of Two Codes, Sci. Rep. 3 - 7493, PML, August 1961

6. L. Calabi and H.G. Haefeli, A Class of Binary Systematic Codes Correcting Errors at Random and In Bursts, IRE Trans IT, Special Supplement, May 1959

Parke Mathematical Laboratories, Inc.
Carlisle, Massachusetts

DECODING RULES FOR CERTAIN PRODUCT CODES

by L.Calabi and H.G. Haefeli (Subcontractor)
March 1963, 25pp. (Scientific Report No. 6;
AFCRL-63-137) (Contract AF19(604)-7493)
Unclassified Report

Decoding procedures are formulated, and their performance established, for products of binary group codes of even minimal weight and for Hobbs' codes of any dimension.

UNCLASSIFIED

1. Information Theory
2. Linear Algebra

I. Calabi, L. and H.G. Haefeli, (Subcontractor)
II. Air Force Cambridge Research Laboratories, Office of Aerospace Research
III. Contract AF19(604)-7493

UNCLASSIFIED

Parke Mathematical Laboratories, Inc.
Carlisle, Massachusetts

DECODING RULES FOR CERTAIN PRODUCT CODES

by L.Calabi and H.G. Haefeli (Subcontractor)
March 1963, 25pp. (Scientific Report No. 6;
AFCRL-63-137) (Contract AF19(604)-7493)
Unclassified Report

Decoding procedures are formulated, and their performance established, for products of binary group codes of even minimal weight and for Hobbs' codes of any dimension.

UNCLASSIFIED

1. Information Theory
2. Linear Algebra

I. Calabi, L. and H.G. Haefeli, (Subcontractor)
II. Air Force Cambridge Research Laboratories, Office of Aerospace Research
III. Contract AF19(604)-7493

UNCLASSIFIED

Parke Mathematical Laboratories, Inc.
Carlisle, Massachusetts

DECODING RULES FOR CERTAIN PRODUCT CODES

by L.Calabi and H.G. Haefeli (Subcontractor)
March 1963, 25pp. (Scientific Report No. 6;
AFCRL-63-137) (Contract AF19(604)-7493)
Unclassified Report

Decoding procedures are formulated, and their performance established, for products of binary group codes of even minimal weight and for Hobbs' codes of any dimension.

UNCLASSIFIED

1. Information Theory
2. Linear Algebra

I. Calabi, L. and H.G. Haefeli, (Subcontractor)
II. Air Force Cambridge Research Laboratories, Office of Aerospace Research
III. Contract AF19(604)-7493

UNCLASSIFIED

Parke Mathematical Laboratories, Inc.
Carlisle, Massachusetts

DECODING RULES FOR CERTAIN PRODUCT CODES

by L.Calabi and H.G. Haefeli (Subcontractor)
March 1963, 25pp. (Scientific Report No. 6;
AFCRL-63-137) (Contract AF19(604)-7493)
Unclassified Report

Decoding procedures are formulated, and their performance established, for products of binary group codes of even minimal weight and for Hobbs' codes of any dimension.

UNCLASSIFIED

1. Information Theory
2. Linear Algebra

I. Calabi, L. and H.G. Haefeli, (Subcontractor)
II. Air Force Cambridge Research Laboratories, Office of Aerospace Research
III. Contract AF19(604)-7493

UNCLASSIFIED